Express Card-FRM11 API Programming (Rev 2.0)

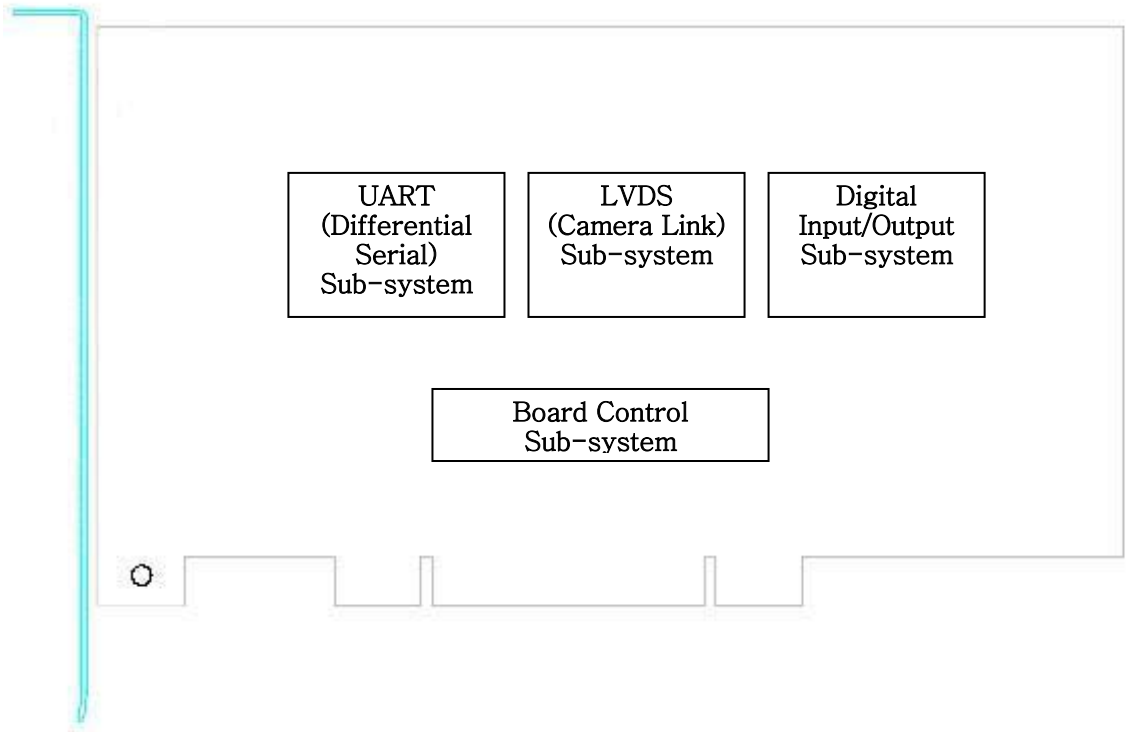
Windows, **Windows2000**, **Windows NT** and **Windows XP** are trademarks of **Microsoft**. We acknowledge that the trademarks or service names of all other organizations mentioned in this document as their own property.

Information furnished by DAQ system is believed to be accurate and reliable. However, no responsibility is assumed by DAQ system for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or copyrights of DAQ system.

The information in this document is subject to change without notice and no part of this document may be copied or reproduced without the prior written consent.

Copyrights © 2005 DAQ system, All rights reserved.

Board Outline and Functional sub-systems



<Functional blocks eCARD-FRM11 board>

We can separate the board function into four functional sub-systems according to the software control view point.

Board Control Sub-system

Controls over-all board operation.

UART Input Sub-system

Controls the functions related with analog input operation.

LVDS Sub-system

Controls the functions related with analog output operation.

Digital Input/Output Sub-system

Controls the functions related with digital input/output operation.

[note] To get more information on each sub-systems capability, please refer to the PCI-FRM01 manual.

© 2005 DAQ system, all rights reserved.

Board Level APIs

Overview

int **OpenDAQDevice**(void)

BOOL **ResetBoard**(int nBoard)

BOOL **CloseDAQDevice**(void)

int **GetBoardNum**(void)

OpenDAQDevice

int **OpenDAQDevice**(void)

Parameters:

None

Return Value:

If the function succeeds, it returns the number of boards which were detected.

If the function fails, the return value is -1, it means there is no device in the system.

ResetBoard

BOOL **ResetBoard**(int nBoard)

Parameters:

nBoard :

The ordinal number of board which you want to initialize. You can get how may boards are installed in this system using **OpenDAQDevice** function.

You may call this function at the very first time you run the program and some suspicious operation.

Return Value:

It returns TRUE in case of the success of reset and initialization.

If you get FALSE you should not call any API functions with the board and call the **CloseDAQDevice()** instead.

CloseDAQDevice

BOOL **CloseDAQDevice**(void)

The CloseDAQDevice function closes all opened devices(boards).

Parameters:

None

© 2005 DAQ system, all rights reserved.

Return Value:

It the function succeeds, it returns TRUE.

GetBoardNum

int **GetBoardNum**(void)

Returns currently detected board number in the system.

Parameters:

None

Return Value:

The number of boards, if succeeds, else returns zero or -1

© 2005 DAQ system, all rights reserved.

UART APIs

Overview

BOOL **UART_Init**(void)

BOOL **UART_GetData**(DWORD* nCnt, unsigned char* buf)

BOOL **UART_SendData**(DWORD* nCnt, unsigned char* buf)

BOOL **UART_Close**(void)

BOOL **UART_SetBaud**(DWORD nBaud)

BOOL **UART_BufferFlush**(void)

UART_Init

Initialize resources used for the UART sub-system, for example interrupt and UART control register

BOOL **UART_Init**(void)

Parameters:

None

Return Value:

If the function succeeds, it returns TRUE.

UART_GetData

Receive characters through the differential UART

BOOL **UART_GetData**(DWORD* nCnt, unsigned char* buf)

Parameters:

nCnt :

The address which contains the number of characters to be received.

The maximum number of characters to be received is limited to 4Kbyte(4096).

buf :

The buffer address.

Return Value:

If the function succeeds, it returns TRUE.

UART _SendData

Sent characters through the differential UART

BOOL **UART_SendData**(DWORD* nCnt, unsigned char* buf)

Parameters:

© 2005 DAQ system, all rights reserved.

nCnt :

The address which contains the number of characters to be sent.

The maximum number of characters to be sent is limited to 4K byte(4096).

buf :

The buffer address.

Return Value:

If the function succeeds, it returns TRUE.

UART _Close

Release all resource were used for UART function.

BOOL **UART_Close**(void)

Parameters:

None

Return Value:

If the function succeeds, it returns TRUE.

UART _SetBaud

Sets UART Baud rates

BOOL **UART_SetBaud**(DWORD nBaud)

Parameters:

nBaud :

0: 9600, 1: 19200, 2: 38400, 3:57600, 4: 115200

Return Value:

If the function succeeds, it returns TRUE.

UART _BufferFlush

Flushes UART RX Buffer

BOOL **UART_BufferFlush**(void)

Parameters:

None

Return Value:

If the function succeeds, it returns TRUE.

© 2005 DAQ system, all rights reserved.

LVDS(Camera Link) APIs

Overview

BOOL **LVDS_Init** (void)

BOOL **LVDS_Start** (void)

BOOL **LVDS_Check** (void)

BOOL **LVDS_GetFrame** (DWORD* nCnt, unsigned char* buf)

BOOL **LVDS_Close** (void)

BOOL **LVDS_Stop** (void)

BOOL **LVDS_SetResolution** (DWORD xRes, DWORD yRes)

BOOL **LVDS_GetResolution** (DWORD *xRes, DWORD *yRes)

BOOL **LVDS_SetDataMode** (int nMode)

BOOL **LVDS_SetFilter** (DWORD dwValue)

BOOL **LVDS_GetFilter** (DWORD *dwValue)

BOOL **LVDS_SetHsCount** (int nCount)

BOOL **LVDS_GetHsCount** (int *nCount)

BOOL **LVDS_GetVersion** (int *nVersion)

LVDS_Init

Initialize resources used for the LVDS sub-system, for example interrupt and LVDS control register

BOOL **LVDS_Init** (void)

Parameters:

None

Return Value:

If the function succeeds, it returns TRUE.

LVDS_Start

Start receiving frame data.

BOOL **LVDS_Start** (void)

Parameters:

None

Return Value:

If the function succeeds, it returns TRUE.

© 2005 DAQ system, all rights reserved.

LVDS_Check

Check the completion of receiving frame data.

BOOL **LVDS_Check** (void)

Parameters:

None

Return Value:

If the frame data were completely received, it returns TRUE.

LVDS_GetFrame

Get Frame data from the frame buffer.

BOOL **LVDS_GetFrame** (DWORD* nCnt, unsigned char* buf)

Parameters:

nCnt :

The address which contains the number of data to be received in byte size.

The maximum number of data to be received is limited to 16Mbyte(0x1000000).

DXD810 use 6MB(0x600000), DXD1417 use 15MB(0xF00000)

buf :

The buffer address.

Return Value:

If the function succeeds, it returns TRUE.

LVDS_Close

Release all resource were used for LVDS function. Application program should call this function when it terminate program

BOOL **LVDS_Close** (void)

Parameters:

None

Return Value:

If the function succeeds, it returns TRUE.

LVDS_SetModel

Set Board interface for specific camera Model interface (Obsolete Function)

BOOL **LVDS_SetModel** (int Model)

Parameters:

Model:

© 2005 DAQ system, all rights reserved.

Model number ‘0’ selects 2048 x 1560 resolution Camera,

otherwise selects 3160 x 2560 resolution Camera.

Return Value:

If the function succeeds, it returns TRUE.

LVDS_GetModel

Gets currently configured camera Model interface (Obsolete Function)

BOOL **LVDS_GetModel** (int *Model)

Parameters:

Model:

Address pointer to receive Camera Model

Return Value:

If the function succeeds, it returns TRUE.

When call succeeds, Model contains currently configured model type.

“0” is 2048 x 1560 resolution camera, others 3160 x 2560 resolution camera.

LVDS_Start

Stop receiving frame data.

BOOL **LVDS_Stop** (void)

Parameters:

None

Return Value:

If the function succeeds, it returns TRUE.

LVDS_SetResolution

Set camera resolution for the specific camera Model.

BOOL **LVDS_SetResolution** (DWORD xRes, DWORD yRes)

Parameters:

xRes:

Value of the horizontal Camera resolution

yRes:

Value of the vertical Camera resolution

Return Value:

If the function succeeds, it returns TRUE.

LVDS_GetResolution

© 2005 DAQ system, all rights reserved.

Gets currently configured camera's frame resolution

BOOL **LVDS_GetResolution** (DWORD *xRes, DWORD *yRes)

Parameters:

xRes:

Address pointer to receive horizontal Camera resolution

yRes:

Address pointer to receive vertical Camera resolution

Return Value:

If the function succeeds, it returns TRUE.

LVDS_SetDataMode

Set image pixel data mode.

BOOL **LVDS_SetDataMode** (int nMode)

Parameters:

nMode:

If the value is 2, the pixel data be expressed by 24bits, others be 16bits.

Return Value:

If the function succeeds, it returns TRUE.

LVDS_SetFilter

Set vsync filter value.

BOOL **LVDS_SetFilter** (DWORD dwValue)

Parameters:

dwValue:

To discriminate real image frame from noisy frames, set the filter value.

The unit of filter value is 15nSEC.

Return Value:

If the function succeeds, it returns TRUE.

LVDS_GetFilter

Get vsync filter value.

BOOL **LVDS_GetFilter** (DWORD *dwValue)

Parameters:

dwValue:

© 2005 DAQ system, all rights reserved.

The address pointer of the filter value which was programmed before.

Return Value:

If the function succeeds, it returns TRUE.

LVDS_SetHsCount

Set horizontal sync count to check real image. (supported by version 2 or more)

BOOL **LVDS_SetHsCount** (int nCount)

Parameters:

nCount:

The minimum value of the horizontal sync of the frame. If the frame has less than the value, the frame data will be thrown away.

Return Value:

If the function succeeds, it returns TRUE.

LVDS_GetHsCount

Get horizontal sync count . (supported by version 2 or more)

BOOL **LVDS_GetHsCount** (int *nCount)

Parameters:

nCount:

The pointer of the horizontal count value.

Return Value:

If the function succeeds, it returns TRUE.

LVDS_GetVersion

Get FPGA version .

BOOL **LVDS_GetVersion** (int *nVersion)

Parameters:

nVersion:

The pointer of the fpga version.

Some API are only supported by boards which have the FPGA version number 2 or more.

Return Value:

If the function succeeds, it returns TRUE.

© 2005 DAQ system, all rights reserved.

Digital Input Output APIs

Overview

unsigned char **DIO_Read** (void)

BOOL **DIO_Write** (unsigned char val)

DIO_Read

Reads from input port

unsigned char **DIO_Read** (void)

Parameters:

None

Return Value:

If the function succeeds, it returns the port value.

If the function fail, it returns the -1(0xff).

DIO_Write

Writes to output port

BOOL **DIO_Write** (unsigned char val)

Parameters:

val :

The value to be written to the port.

Return Value:

If the function succeeds, it returns TRUE.

© 2005 DAQ system, all rights reserved.